Business driven development
March 2006

Rational. software

IBM

Governing the business process of software and systems development.

Liz Barnett, EZ Insight Inc.
Murray Cantor and Rachael Rusting, IBM Corporation

ON DEMAND BUSINESS™

Contents

Introduction

Software runs today's businesses. Therefore, the success of software and systems development projects can directly impact a company's survival. Competition is fierce and customers demand more. It's not surprising that expectations for software quality and usability have risen dramatically. But the current business climate presents even greater challenges due to the confluence of two different factors. First, recent corporate scandals have shown that there are direct legal implications for poor management and compliance. Visions of CEOs on trial are still emblazoned in the minds of many executives and shareholders. Second, the competitive pressures that come from an increasingly global business world are forcing companies to be more flexible in meeting customer needs. If not, they will lose out to competitors around the world who are ready, willing and able to provide better or cheaper solutions. Thomas Friedman, in his best-selling book, *The World Is Flat*, put it succinctly:

"If you want to grow and flourish in a flat world, you better learn how to change and align yourself with it."[1]

Together, these two factors present a paradox. Businesses must become much more agile and responsive to changing threats and competitors, while simultaneously satisfying ever higher standards of accountability and compliance.

Many companies are investing in governance at the business level, but today's pressures demand that they also apply the principles of governance to software and systems development. It may seem that governance would be a hindrance to agile business, but the exact opposite is true. When implemented properly, governance enables companies to execute highly flexible and business driven processes in the context of a structured and compliant environment (see Figure 1). This white paper shows how governing the business process of software and systems development—also known as "business driven development"—will allow your organization to achieve this necessary balance between responsiveness and control.

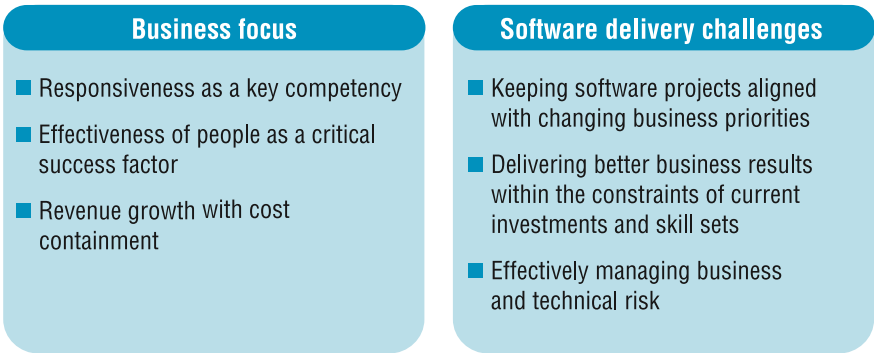

Figure 1. The impact of business challenges on software delivery
Source: IBM Rational research, November 2005

Defining governance

Effective governance is a mandatory component for managing a global enterprise. Governance in business is defined as:

- ***Establishing** chains of responsibility, authority and communication to **empower** people (decision rights).*
- ***Establishing** measurement, policy and control mechanisms to **enable** people to carry out their roles and responsibilities.*

Governance implementations vary widely among companies and are frequently triggered by both internal and external drivers. Some governance programs are focused on complying with specific government and industry regulations, such as Sarbanes-Oxley (SOX) or HIPAA in the United States. Governance programs can also include supporting internal standards and frameworks such as Information Technology Infrastructure Library (ITIL) and Capability Maturity Model Integration (CMMI) as a means to achieve cost containment and overall risk management for IT. Other organizations concentrate on gaining control over and insight into vendors, contractors and business partners that are participating on their projects. And for others, the

need to be more flexible in meeting business requirements demands closer alignment of technology and business teams. The rules of engagement for making the most of individual and cross-functional team productivity are part of any governance initiative.

Governing software and systems development

The principles of governance are universal, but they are typically customized to address different aspects of the business. If your business depends on software and systems, it is essential that the software-based solutions being delivered meet customers' needs and provide value to the business (see Figure 2).

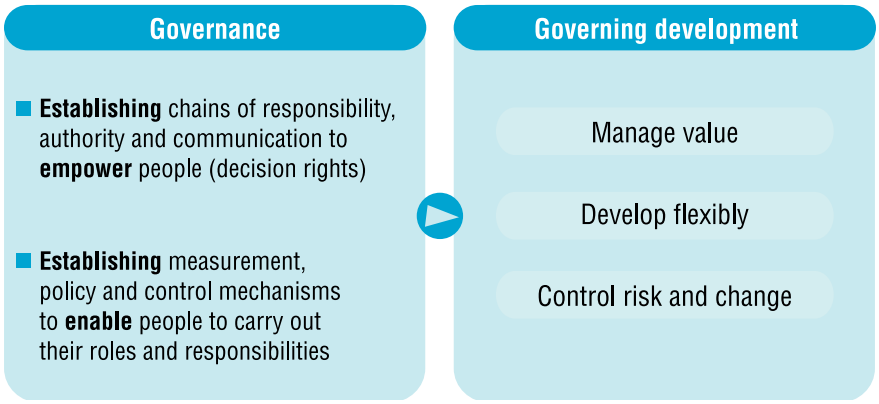

Figure 2. Applying the principles of governance development

Development is no longer limited to creating new software code. In fact, successful organizations want to minimize new code development and leverage existing assets and resources. Development now centers on delivering solutions, which can include:

- *Purchasing or specifying new applications.*
- *Integrating or extending existing applications to support new business processes.*
- *Building software products to be sold to customers and business partners.*

- *Using applications or components of applications from business partners or vendors.*
- *Managing projects that have been outsourced to external suppliers, either onshore or offshore.*
- *Consolidating applications to reduce maintenance costs and redundant systems.*
- *Producing mobile or embedded systems that include both hardware and software components.*

Software projects differ from other types of business projects in that they rely on estimates that have significant variances. Key among these estimates are those that address schedules, costs and effort. Project managers struggle to improve their estimating abilities, yet the risks associated with such variables as new technologies and frameworks, distributed development teams and partnerships with third-party vendors remain beyond their control. If you can't control the sources behind your estimates, the variances can only grow—the greater the variances, the higher the degree of risk. With so many uncontrolled variables and increasing variances, it's necessary to have visibility into and governance of the overall project.

Successfully governing development balances the needs of management and developers. Managers require traceability, predictability and ongoing information about risk. Developers, on the other hand, must be able to collaborate effectively with team members whether local or remote, work in a flexible environment that promotes creativity and quality, and be able to focus and maximize their productivity on specific tasks. Thus, the challenge: companies must provide developers with a highly flexible and agile environment that encourages innovation, and that also provides the transparency needed to manage risk.

Why govern development now?

The age-old dilemma of "faster, better and cheaper" software has plagued organizations for years. Now, businesses are faced with several new challenges that further exacerbate the need for governing software delivery, yet paradoxically provide greater opportunity for development agility (see Figure 3). These challenges include:

- *Geographically distributed development.*
- *Regulatory compliance.*
- *Open computing.*
- *Modular systems, or service-oriented architecture (SOA).*

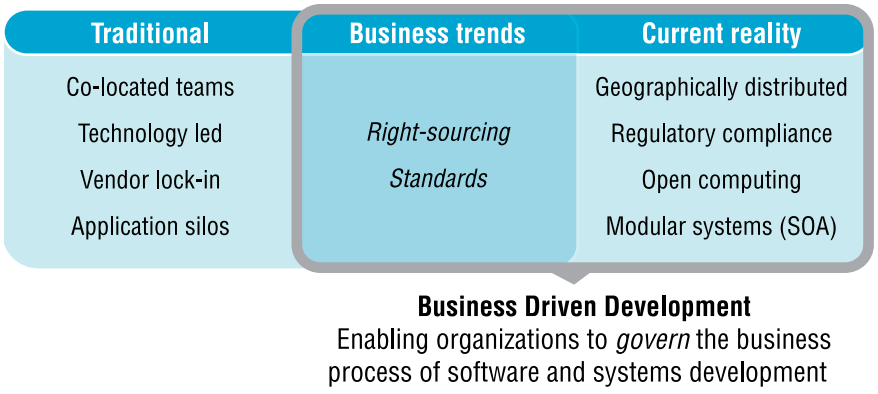

Figure 3. Trends in right-sourcing and standards are creating new challenges and opportunities for software and systems delivery.

Geographically distributed development

The challenge: Global economics and emerging technologies demand that organizations distribute and integrate skills, activities, teams and resources in new and cost-effective ways. But distributed organizations are different in every company. The term *geographically distributed development* can simply refer to the number of physical locations where technology and business

people work. Or it can refer to managing third-party consultants, vendors and partners and their contractual obligations and service-level agreements. Or it can describe the blending of cultures across nations, regional areas and styles (such as development processes) that must be integrated for a single project.

Use of offshore outsourcing, especially in large companies, is on the rise. A recent *CIO* magazine survey noted that only 27 percent of executives surveyed do not expect to outsource beyond the U.S.[2] But moving development offshore doesn't necessarily equate to significant cost reduction. While labor rates may be as much as a 70 percent lower, there will be increased transaction expenses (including legal fees and vendor selection processes) and monitoring costs (including travel and telecommunications). The resulting development environment may only save the company 30 percent—and that is only if it is able to successfully manage the distributed project.[3]

The opportunity: Today, any business activity or process that can be digitized can be sourced to the most knowledgeable workers, no matter what company they work for or where they are located in the world. Right-sourcing means that you draw on the skills of people within and outside your company and put the right people on the right jobs. Therefore, geographically distributed development raises a whole new set of challenges: Who is responsible for what? How are they progressing? How can project assets and information be shared? How can productivity be maximized across geographical and cultural boundaries? An effective governance program must address these questions and provide visibility into all aspects of the geographically distributed development environment.

Regulatory compliance

The challenge: To achieve regulatory compliance, an organization must have documented how its governance program was implemented and how specific requirements were satisfied. Compliance requires governance, but governance alone is not sufficient to demonstrate compliance unless specific goals and measurements related to the regulations have been put in place and fulfilled (see Figure 4).

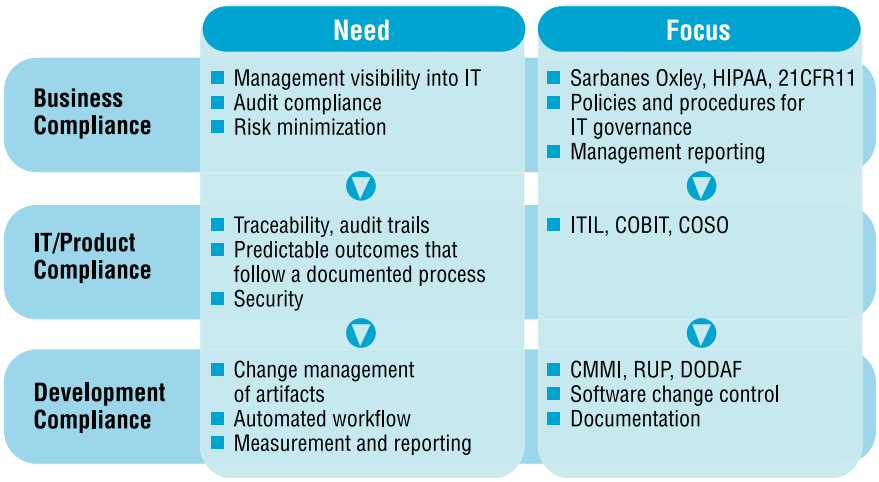

Figure 4. Compliance addresses a wide range of requirements and impacts all levels of the organization.

Organizations must find ways to ensure that their processes and products comply with a variety of government regulations (e.g., SOX, Basel II, 21 CFR Part 11 and HIPAA) in every country in which they operate. Specifically, the compliance component of any IT governance program must ensure traceability and visibility across a secure, distributed and tamper-resistant software development environment. This requirement can be a huge challenge. For example, in a June 2005 *CFO* magazine survey of 153 senior executives, one-third of those responding reported failures in their SOX audits. And of those companies, fully 96 percent attributed these failures to the IT organization.[4]

The necessity: With compliant development environments, executives can manage risk and project teams can have more control and predictability across their projects. This implies an ongoing program to respond quickly to changing domestic and international regulations.

Open computing

The challenge: Open computing is designed to break down walls between technologies, processes and people. It allows teams to collaborate and innovate more effectively. The term "open computing" encompasses open standards for

interoperability and data sharing, open architectures to enable loosely coupled and reconfigurable systems, and open source software to promote community development and collaboration (see Figure 5). But with these increased choices comes increased complexity, so software teams must be equipped to manage these new environments in conjunction with existing tools and techniques.

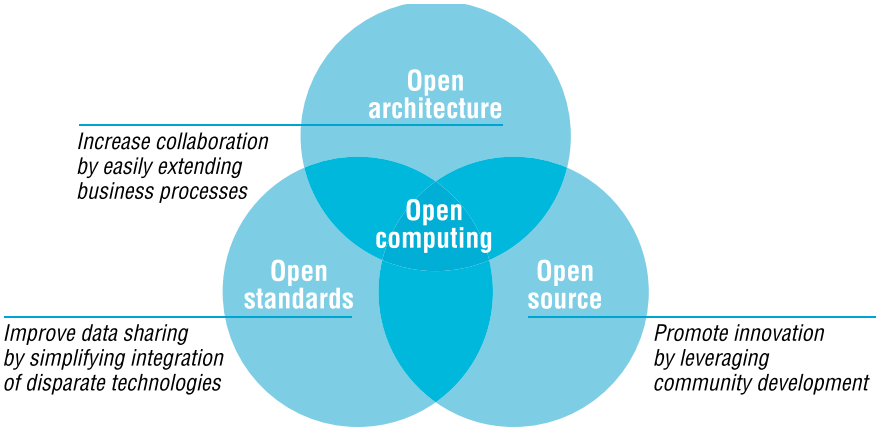

Figure 5. Developers can draw upon a mix of open architectural, standards-based and collaborative techniques.

One of the most difficult questions is: How can the development team leverage open source development experiences within a governance program, yet stick to a schedule that meets business needs? It's important to be able to move from closed, proprietary infrastructures to open, standardized platforms and still maintain control and traceability over your development assets.

The opportunity: Open computing can benefit development in a number of ways. Communities that have formed to develop open standards for software development have been far more effective than standards bodies of the past. Open source projects provide excellent models for strong technical governance, effective collaboration among highly distributed teams, and adherence to a disciplined yet agile process. Developers who have participated in open

source projects, either as part of their professional jobs or voluntarily, understand how this balance of discipline and flexibility can lead to truly impressive software solutions. Many organizations have begun adopting open source project techniques and tools into their own development environments, hoping to capitalize on the success of this collaborative development approach.

Modular systems (SOA)

The challenge: Modular systems—largely implemented as service-oriented architectures (SOAs)—enable businesses to "deconstruct" critical business processes into modular components that can then be "reconstructed" into new processes. Companies are adopting SOAs as a means to quickly deliver value to the business by leveraging the assets found in existing applications or those provided by business partners. This new value may come in the form of new uses of existing systems, business opportunities in new or related markets, cost savings due to consolidation, sharing of common processes, and so on.

SOA applications are intrinsically more fragmented and thus introduce their own set of governance challenges. Resolving traceability, ownership, quality of service and security across all services becomes more difficult to accomplish. The resulting solution will be, at best, no stronger in these areas than the weakest of its component parts.

The opportunity: Development organizations must determine how to manage the many collections of independent and loosely coupled services that deliver a business function. In addition, partner-provided services need to satisfy internal governance requirements. Taking advantage of existing internal and partner-provided services and effectively managing those business functions can make the difference between success and failure in delivering results in a competitive market.

Implementing governance for development

For successful business driven development, a governance program must provide a framework to manage value, develop flexibly and control risk and change (see Figure 6). Each of these three dimensions requires changes in the way that many development organizations are managed today.

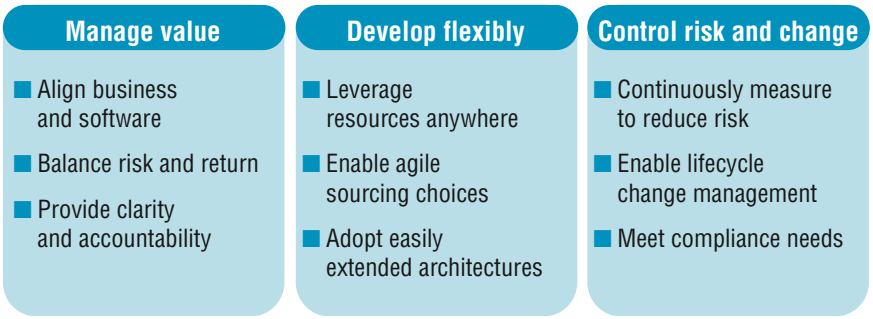

Figure 6. Organizations govern development as a business process by managing value, developing flexibly, and controlling risk and change.

Managing value

One of the most challenging aspects of a governance program for software and systems development is managing change. This challenge is especially true for organizations that have historically been managed as cost centers. Few, if any, development organizations have been accountable for measuring, let alone assessing, the change that they deliver. Software projects must be directly aligned with business goals, and so they must undergo the same type of risk and return analysis that is found in any other type of business initiative. To do this, the project leaders, both technical and management, must be proficient in the company's business processes and be able to tie their activities back to new or ongoing programs. Expected value should be determined at the onset of a project, when the team evaluates initiatives alongside business priorities. Then, value must be managed and tracked throughout the life of the project.

It is essential to manage value at both the project and the organizational level across the lifecycle of a solution. At the project level, this means ensuring a project's ability to meet business needs using available resources, and having transparency into the project's health and status. At the organizational level, it means managing resources dynamically and flexibly across a portfolio of projects while ensuring compliance with internal, business and/or regulatory requirements.

Developing flexibly

Business agility requires development organizations to quickly and cost-effectively deliver new software capabilities and solutions. Rapid innovation that delivers value to the business is critical. Development teams need to employ a mix of strategies including:

- *Staffing distributed teams that leverage resources across geographic and organizational boundaries.*
- *Leveraging existing technology assets and alternative sources to improve productivity and quality.*
- *Adopting more agile and iterative techniques and easily extended architectures such as SOA.*

Controlling risk and change

Over the last forty years, we have learned that it is impossible to achieve full understanding of requirements at the onset of all but the most routine projects. Hence it is impossible to make highly accurate estimates of the effort to meet these requirements. At the inception of any development project, there is always uncertainty, particularly in areas such as effort, cost and time to delivery. Estimating these project parameters is critical to a development project's success, but we know that these parameters always have some amount of statistical variance. In fact, the variances in the estimates of these types of project parameters are typically quite high. As the variance in the estimates increase, the likelihood of successfully completing the project decreases.

The main reason development projects fail to meet stakeholder needs is that they are governed as if these variances do not exist. A simple approach to addressing schedule and budget variances is to push out the project schedule and thus increase the budget. But this is not practical or effective for most projects. Budgets continue to decrease and the business will not wait. A better approach is to acknowledge these uncertainties and manage the project so that uncertainties and risks are removed early in the project, increasing the team's ability to make accurate and aggressive plans.

The Risk Workoff curve (see Figure 7) provides a means of collecting a wide range of metrics and then tracking reductions in variances over time. Metrics for governance should include, but not be limited to, the following types of measurements:

- *Progress achieved*
- *Risk removed (such as reduction of variance of key project metrics)*
- *Assessment of other project targets including quality and financial requirements*

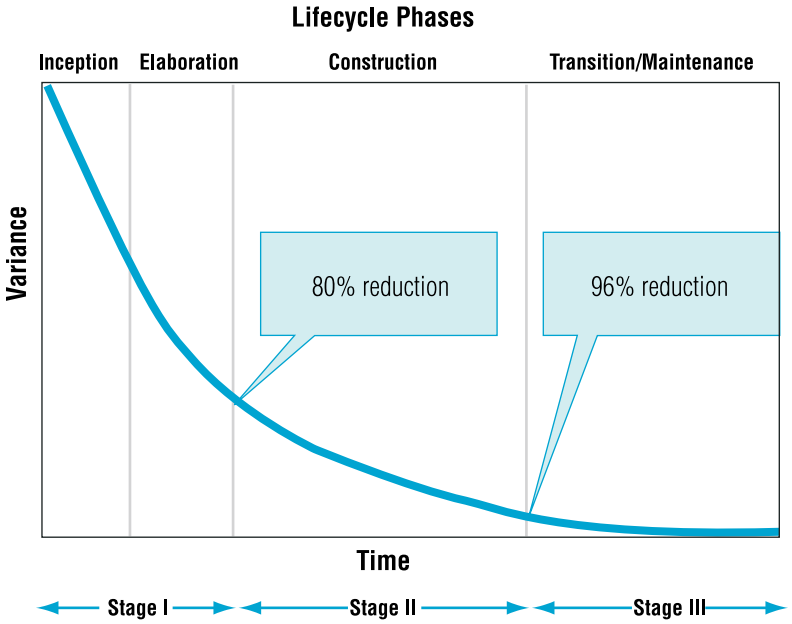

Figure 7. Addressing the range of variances early in a project's lifecycle can significantly decrease a project's overall risk.

Developing flexibly using an iterative development process provides a framework for this approach. As the team progresses through each iteration, it has the ability to reassess the project status in light of increased knowledge and decreased uncertainty, and it can adjust the project estimates accordingly. Teams that remove risk early in the project's iterations by refining scope, validating technical architecture or other adjustments can improve the chances that later iterations and the project as a whole are successful.

Getting started now

How do you know if you need to better govern your software and systems development projects? You can start by asking the following questions.

Are you able to:

- *Consistently align software systems and projects with business goals?*
- *Understand the real cost of delivering a software-based system?*
- *Leverage geographically distributed development resources?*
- *Comply with regulatory standards?*
- *Quickly extend business processes to include third parties and partners?*
- *Gain real-time visibility into projects without creating overhead?*
- *Enable software teams to innovate, yet be accountable?*

Companies that cannot answer "yes" to all of the above questions have weak governance programs. When development projects are not aligned with business goals, the business sees the development organization as being unresponsive. When software or systems solutions do not deliver value to the business, the CIO is viewed as running an ineffective cost center. And when management feels as if it has lost control of a project because it lacks current or valid information, it generally assumes that developers are hiding issues and even failing. Many of these misconceptions come down to the fact that developers are not managing risk, quantifying the value that they deliver and communicating that information effectively.

So where do you start? At the core of a governance program, there are three areas that every development organization must determine:

1. *What must be addressed or fixed?*
2. *What must be controlled?*
3. *What must be measured to determine and communicate success?*

Many development organizations don't stop to analyze exactly what it is that they need to do in order to govern development. It may be as specific as the need to comply with a government or industry regulation. Or it may be that quality or time-to-market issues are directly affecting their ability to compete. This first step of assessing your organization's current status, open issues and necessary requirements must be completed before you invest any time in governance.

Second, you can't control what you can't measure. To address control in a development environment, you must proactively identify risks and areas where variances are high. At the program or project level, it's important to focus on project schedules and effort, software quality, technology factors and other high risk areas that can affect a manager's ability to estimate a project. At the organizational level, you need to look at risks associated with a broad portfolio of projects, and addressing areas such as real-time resource management, project and program prioritization, risk and return analysis, and the direct alignment of development projects to specific business initiatives.

And finally, measurement gives you the ability to assess status and make adjustments to improve on a particular area. Measurement also affects behavior, so you must be sure that the things that you track will lead to the behaviors that you want. Management indicators (such as those that track progress, effort, cost and resource management) should be extended to include metrics that address value and risk. Too often, measurements appear as a simple number as opposed to a distribution that gives a better sense of the possible range of values. As discussed in this paper, risk is essentially variance of estimates. By understanding that every estimate is just one value in a distribution, you get some sense for what the range of possible outcomes might be.

Management metrics should be supplemented by quality indicators that address software changes (such as volume and frequency), rework and breakage to software and systems. For example, companies have begun introducing customer satisfaction metrics into their development programs to help determine if and how development teams are delivering value to the business. Depending on the problem being addressed and the parameters necessary to control, a different set of metrics may be appropriate.

IBM helps development organizations implement effective governance programs. An initial assessment can identify high value opportunities, pain points and potential pilot projects. IBM's extensive resources can help educate your development staff and assist with skills transfer. And we will work with you to design and implement a governance program to address your local and global development initiatives.

To learn more

Contact the IBM Rational® team at 1 800 728-1212 or send us an e-mail at ratlinfo@us.ibm.com, for more information on how to achieve business driven development.

IBM Corporation
IBM Software Group
Route 100
Somers, NY 10589
U.S.A.

Produced in the United States of America
03-06

1 *The World Is Flat*, Thomas L. Friedman, p. 339

2 *CIO* magazine, "The State of the CIO 06" survey. See "http://www.cio.com/archive/010106/J AN1SOC.pdf" http://www.cio.com/archive/010106/JAN1SOC.pdf

3 "Offshore Outsourcing—Business Models, ROI and Best Practices" Marcia Robinson and Ravi Kalakota

4 *CFO* magazine, http://www.cfo.com/article.cfm/4077489

G507-1915-00